PORTAL USPTO
Subscribe (Full Service) Register (Limited Service, Free) Login
Search: The ACM Digital Library The Guide
tree node insert SEARCH
THE ACM DIGITAL LIBRARY
Feedback Report a problem Satisfaction survey

Terms used **tree node insert** Found **34,141** of **184,245**

Sort results by: relevance
Display results: expanded form
Save results to a Binder
Search Tips
Open results in a new window
Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next
Best 200 shown Relevance scale

1 Routing: An optimal jumper insertion algorithm for antenna avoidance/fixing on general routing trees with obstacles
Bor-Yiing Su, Yao-Wen Chang, Jiang Hu
April 2006 **Proceedings of the 2006 international symposium on Physical design ISPD '06**
Publisher: ACM Press
Full text available: pdf(304.28 KB) Additional Information: full citation, abstract, references, index terms

We study in this paper the problem of jumper insertion on general routing (Steiner/spanning) trees with obstacles for antenna avoidance/fixing at the routing and/or post-layout stages. We formulate the jumper insertion for antenna avoidance/fixing as a tree-cutting problem and present the *first*optimal algorithm for the *general* tree-cutting problem. We show that the tree-cutting problem exhibits the properties of optimal substructures and greedy choices. With these properties, we pr ...

Keywords: antenna effect, jumper insertion

2 Utilization of B-trees with inserts, deletes and modifies
T. Johnson, D. Shasha
March 1989 **Proceedings of the eighth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems**
Publisher: ACM Press
Full text available: pdf(1.07 MB) Additional Information: full citation, abstract, references, citings, index terms

The utilization of B-tree nodes determines the number of levels in the B-tree and hence its performance. Until now, the only analytical aid to the determination of a B-tree's utilization has been the analysis by Yao and related work. Yao showed that the utilization of B-tree nodes under pure inserts was 69%. We derive analytically and verify by simulation the utilization of B-tree nodes constructed from N inserts followed by M modifies (where M

3 Industrial sessions: the marriage of XML and relational databases: ORDPATHs: insert-friendly XML node labels
Patrick O'Neil, Elizabeth O'Neil, Shankar Pal, Istvan Cseri, Gideon Schaller, Nigel Westbury
June 2004 **Proceedings of the 2004 ACM SIGMOD international conference on Management of data**
Publisher: ACM Press
Full text available: pdf(101.14 KB) Additional Information: full citation, abstract, references

We introduce a hierarchical labeling scheme called ORDPATH that is implemented in the upcoming version of Microsoft® SQL Server™. ORDPATH labels nodes of an XML tree without requiring a schema (the most general case---a schema simplifies the problem).

An example of an ORDPATH value display format is "1.5.3.9.1". A compressed binary representation of ORDPATH provides document order by simple byte-by-byte comparison and ancestry relationship equally simply. In addition, the ORDPATH schem ...

4 An empirical study of insertion and deletion in binary search trees

Jeffrey L. Eppinger
September 1983 **Communications of the ACM**, Volume 26 Issue 9
Publisher: ACM Press
Full text available: pdf(655.83 KB) Additional Information: full citation, abstract, references, citings, index terms

This paper describes an experiment on the effect of insertions and deletions on the path length of unbalanced binary search trees. Repeatedly inserting and deleting nodes in a random binary tree yields a tree that is no longer random. The expected internal path length differs when different deletion algorithms are used. Previous empirical studies indicated that expected internal path length tends to decrease after repeated insertions and asymmetric deletions. This study shows that p ...

Keywords: binary trees

5 An optimal insertion algorithm for one-sided height-balanced binary search trees

Kari-Jouko Räihä, Stuart H. Zweben
September 1979 **Communications of the ACM**, Volume 22 Issue 9
Publisher: ACM Press
Full text available: pdf(438.98 KB) Additional Information: full citation, abstract, references, citings

An algorithm for inserting an element into a one-sided height-balanced (OSHB) binary search tree is presented. The algorithm operates in time O(log n), where n is the number of nodes in the tree. This represents an improvement over the best previously known insertion algorithms of Hirschberg and Kosaraju, which require time O(log2n). Moreover, the O(log n
Keywords: binary trees, height-balanced trees, insertion, one-sided height-balanced trees, search trees

6 An insertion algorithm for a minimal internal path length binary search tree

Thomas E. Gerasch
May 1988 **Communications of the ACM**, Volume 31 Issue 5
Publisher: ACM Press
Full text available: pdf(1.49 MB) Additional Information: full citation, abstract, references, citings, index terms, review

This paper presents an insertion algorithm for maintaining a binary search tree with minimal internal path length. The insertion algorithm maintains minimal internal path length by displacing keys when necessary, in an inorder fashion, until a vacant position is found in the last incomplete level of the tree. The algorithm produces trees that are optimal for searching while exhibiting a runtime behavior that is between logarithmic and linear in the number of nodes in the tree, with linear t ...

7 A Cost Model for the Internal Organization of B+-Tree Nodes

Wilfred J. Hansen
October 1981 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 3 Issue 4
Publisher: ACM Press
Full text available: pdf(1.43 MB) Additional Information: full citation, references, citings, index terms

8 Multimedia and visualization: Node splitting algorithms in tree-structured high-

dimensional indexes for similarity search
Yongjian Fu, Jui-Che Teng, S. R. Subramanya
March 2002 **Proceedings of the 2002 ACM symposium on Applied computing**
Publisher: ACM Press
Full text available: pdf(462.00 KB) Additional Information: full citation, abstract, references, index terms

Content-based searches and retrievals in multimedia and image databases use high-dimensional indexing structures for organizing the features of the objects. Most of those index structures are tree-structured whose nodes have a limit on the number of entries describing the subtrees rooted at those nodes. When index trees are built by repeated insertion of entries, nodes need to be split and the tree balanced accordingly. Node-splitting algorithms eventually determine the final structure of the tr ...

Keywords: content-based retrieval, high-dimensional index, multimedia / image database, node splitting, similarity search

9 The hB $^\Pi$-tree: a multi-attribute index supporting concurrency, recovery and node consolidation
Georgios Evangelidis, David Lomet, Betty Salzberg
February 1997 **The VLDB Journal — The International Journal on Very Large Data Bases**, Volume 6 Issue 1
Publisher: Springer-Verlag New York, Inc.
Full text available: pdf(314.15 KB) Additional Information: full citation, abstract, citings, index terms

We propose a new multi-attribute index. Our approach combines the hB-tree, a multi-attribute index, and the Π-tree, an abstract index which offers efficient concurrency and recovery methods. We call the resulting method the hB $^\Pi$-tree. We describe several versions of the hB $^\Pi$-tree, each using a different node-splitting and index-term-posting algorithm. We also describe a new node deletion algorithm. We have implemented all the versions of the hB $^\Pi$-tree. Our performance results ...

Keywords: Concurrency, Multi-attribute index, Node consolidation, Recovery

10 An Algorithm for Zero-Skew Clock Tree Routing with Buffer Insertion
Y. P. Chen, D. F. Wong
March 1996 **Proceedings of the 1996 European conference on Design and Test**
Publisher: IEEE Computer Society
Full text available: pdf(824.16 KB) Publisher Site Additional Information: full citation, abstract, citings

We study the problem of multi-stage zero skew clock tree construction for minimizing clock phase delay and wirelength. In existing approaches clock buffers are inserted only after clock tree is constructed. The novelty of this paper lies in simultaneously perform clock tree routing and buffer insertion. We propose a clustering-based algorithm which uses shortest delay as the cost function. We show that the feasible positions for clock tree nodes and buffers can be generalized from diagonal segme ...

Keywords: clock skew, clock routing, buffer, phase delay, shortest path

11 Tree insertion grammar: a cubic-time, parsable formalism that lexicalizes context-free grammar without changing the trees produced
Yves Schabes, Richard C. Waters
December 1995 **Fuzzy Sets and Systems , Computational Linguistics**, Volume 76 , 21 Issue 3 , 4
Publisher: Elsevier North-Holland, Inc. , MIT Press
Full text available: pdf(2.31 MB) Additional Information: full citation, abstract, references

Publisher Site

Tree insertion grammar (TIG) is a tree-based formalism that makes use of tree substitution and tree adjunction. TIG is related to tree adjoining grammar. However, the adjunction permitted in TIG is sufficiently restricted that TIGs only derive context-free languages and TIGs have the same cubic-time worst-case complexity bounds for recognition and parsing as context-free grammars. An efficient Earley-style parser for TIGs is presented.Any context-free grammar (CFG) can be converted into a lexica ...

12 An immediate approach to balancing nodes in binary search trees

Chung-Chih Li

April 2006 **Journal of Computing Sciences in Colleges**, Volume 21 Issue 4

Publisher: Consortium for Computing Sciences in Colleges

Full text available: pdf(534.14 KB) Additional Information: full citation, abstract, references, index terms

We present an immediate approach in hoping to bridge the gap between the difficulties of learning ordinary Binary Search Trees (henceforth BST) and their height-balanced variants in the typical data structures class. For each node in the tree, instead of balancing the heights of the two children, we balance the number of their nodes. The concepts and programming techniques required are much easier than the AVL tree or any other height-balanced alternatives. We also show that the height of a node ...

Keywords: binary search trees, data structures, node-balanced binary search trees

13 Power trees

Fabrizio Luccio, Linda Pagli

November 1978 **Communications of the ACM**, Volume 21 Issue 11

Publisher: ACM Press

Full text available: pdf(590.42 KB) Additional Information: full citation, abstract, references, citings, index terms

The new class of Pk trees is presented, where height balance is maintained for the nodes lying on particular paths. The number of nodes of a Pk tree asympotically grows as a power of the height, in the worst case. A procedure for node insertion is given, and the class of trees considered is restricted to IPk trees, which are buildable by such a procedure. The average behavior of such trees, studied by an extensive set of simulation runs, is close to that of AVL trees. In particular, the fam ...

Keywords: IPk trees, binary search trees, node insertion, search length, subtree rotation

14 Finger search trees with constant insertion time

Gerth Stølting Brodal

January 1998 **Proceedings of the ninth annual ACM-SIAM symposium on Discrete algorithms**

Publisher: Society for Industrial and Applied Mathematics

Full text available: pdf(1.16 MB) Additional Information: full citation, references, citings, index terms

15 An efficient R-tree implementation over flash-memory storage systems

Chin-Hsien Wu, Li-Pin Chang, Tei-Wei Kuo

November 2003 **Proceedings of the 11th ACM international symposium on Advances in geographic information systems GIS '03**

Publisher: ACM Press

Full text available: pdf(190.75 KB) Additional Information: full citation, abstract, references, citings, index terms

For many applications with spatial data management such as Geographic Information

Systems (GIS), block-oriented access over flash memory could introduce a significant number of node updates. Such node updates could result in a large number of out-place updates and garbage collection over flash memory and damage its reliability. In this paper, we propose a very different approach which could efficiently handle fine-grained updates due to R-tree index access of spatial data over flash memory. The ...

Keywords: GIS, R-tree, embedded systems, flash memory, spatial index structures, storage systems

16 Efficient locking for concurrent operations on B-trees

Philip L. Lehman, s. Bing Yao
December 1981 **ACM Transactions on Database Systems (TODS)**, Volume 6 Issue 4
Publisher: ACM Press
Full text available: pdf(1.40 MB) Additional Information: full citation, abstract, references, citings, index terms

The B-tree and its variants have been found to be highly useful (both theoretically and in practice) for storing large amounts of information, especially on secondary storage devices. We examine the problem of overcoming the inherent difficulty of concurrent operations on such structures, using a practical storage model. A single additional "link" pointer in each node allows a process to easily recover from tree modifications performed by other concurrent processes. Our solution ...

Keywords: B-tree, concurrenct algorithms, concurrency controls, consistencey, correctness, data structures, database, index organizations, locking protocols, multiway search trees

17 Fringe analysis of binary search trees with miniml internal path length

Faris N. Abuali, Roger L. Wainwright
April 1991 **Proceedings of the 19th annual conference on Computer Science**
Publisher: ACM Press
Full text available: pdf(662.26 KB) Additional Information: full citation, references

18 Scapegoat trees

Igal Galperin, Ronald L. Rivest
January 1993 **Proceedings of the fourth annual ACM-SIAM Symposium on Discrete algorithms**
Publisher: Society for Industrial and Applied Mathematics
Full text available: pdf(1.02 MB) Additional Information: full citation, references, citings, index terms

19 XML indexing and compression: ViST: a dynamic index method for querying XML data by tree structures

Haixun Wang, Sanghyun Park, Wei Fan, Philip S. Yu
June 2003 **Proceedings of the 2003 ACM SIGMOD international conference on Management of data**
Publisher: ACM Press
Full text available: pdf(244.47 KB) Additional Information: full citation, abstract, references, citings, index terms

With the growing importance of XML in data exchange, much research has been done in providing flexible query facilities to extract data from structured XML documents. In this paper, we propose ViST, a novel index structure for searching XML documents. By representing both XML documents and XML queries in structure-encoded sequences, we show that querying XML data is equivalent to finding subsequence matches. Unlike index methods that disassemble a query into multiple sub-queries, and then *joi* ...

20 Statistical optimization and manufacturability: An exact jumper insertion algorithm for antenna effect avoidance/fixing

Bor-Yiing Su, Yao-Wen Chang

June 2005 **Proceedings of the 42nd annual conference on Design automation**

Publisher: ACM Press

Full text available: pdf(538.05 KB) Additional Information: full citation, abstract, references, index terms

As the process technology enters the nanometer era, reliability has become a major concern in the design and manufacturing of VLSI circuits. In this paper we focus on one reliability issue-jumper insertion in routing trees for avoiding/fixing antenna effect violations at the routing/post-layout stages. We formulate the jumper insertion for antenna avoidance/fixing as a tree-cutting problem. We show that the tree-cutting problem exhibits the properties of optimal substructures and greedy choices. ...

Keywords: antenna effect, jumper insertion

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2006 ACM, Inc.
Terms of Usage Privacy Policy Code of Ethics Contact Us

Useful downloads: Adobe Acrobat QuickTime Windows Media Player Real Player